UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/625,908	07/24/2003	Axel Von Bergen	13909-119001 / 2003P00132	1208

32864 7590 06/02/2006

FISH & RICHARDSON, P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
BRADLEY, MATTHEW A	

ART UNIT	PAPER NUMBER
2187	

DATE MAILED: 06/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/625,908	Applicant(s) BERGEN ET AL.	
	Examiner Matthew Bradley	Art Unit 2187	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 March 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

This Office Action has been issued in response to amendment filed 16 March 2006. Applicant's arguments have been carefully and fully considered in light of the instant amendment, but they are not persuasive. Accordingly, this action has been made FINAL.

Claim Status

Claims 1-24 remain pending and are ready for examination.

Claim Objections

The objections to claim 1 set forth in the Office Action dated 16 December 2005 have been withdrawn in light of the instant amendment.

Double Patenting

The double patenting rejection set forth in the Office Action dated 16 December 2005 has been withdrawn in light of the terminal disclaimer filed.

Claim Rejections - 35 USC § 101

The 35 USC 101 rejections set forth in the Office Action dated 16 December 2005 have been withdrawn in light of the instant amendment.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-24 are rejected under 35 U.S.C. 102(b) as being anticipated by McMahon et al. (U.S. 5,784,699) herein after referred to as McMahon.

As per independent claim **1**, McMahon teach,

- outputting a request from an application to an operating system for allocation of a block of memory by the operating system to the application; (Column 5 lines 25-27)
- accessing the block of memory for the application; (Column 5 lines 30-39)
- dividing the block of memory into frames; (Column 4 lines 37-40)
- dividing each of the frames into instances; and associating an application-defined instance type with the instances for data storage using the instances. (Column 5 lines 50-59).
    - *The Examiner notes that the dynamic memory allocator takes the blocks (frames) from the whole memory and subsequently divides the blocks (frames) into portions (instances) for allocations. As the request for memory from the application, comes from the application, any allocated memory for that application will contain space available for that application to use. Accordingly, the allocation of memory for the application will contain data storage for the instances using the instances of the requesting application.*

As per dependent claim **2**, McMahon teach, allocating a block of memory that begins on a page boundary (Column 3 lines 28-37). *The Examiner notes that a free list is present in the system of McMahon. As shown in the table in Column 6 of McMahon,*

the dynamic memory allocator, in the form of these free lists, catalogs various free blocks of memory by size. Accordingly, when an allocation takes place by the dynamic memory allocator, for example of the 16-byte size, the memory allocated will begin on the 16-byte boundary.

As per dependent claim **3**, McMahon teach, wherein the size of the block of memory is determined by a coding parameter associated with the application (Column 3 lines 6-8).

As per dependent claim **4**, McMahon teach, wherein dividing the block of memory into frames includes identifying a first page boundary within the block of memory (Column 3 lines 28-37). *The Examiner notes that a free list is present in the system of McMahon. As shown in the table in Column 6 of McMahon, the dynamic memory allocator, in the form of these free lists, catalogs various free blocks of memory by size thereby indicating a boundary for each size that is maintained.*

As per dependent claim **5**, McMahon teach, wherein dividing the block of memory into frames further includes designating a portion of the block of memory before the first page boundary as unused (Column 6 lines 34-37). *The Examiner notes that as discussed supra and with respect to the instant citation, the allocator is able to divide the memory block and release the remainder of the block as free space. Accordingly, the allocator designates a portion of the block of memory as unused.*

As per dependent claim **6**, McMahon teach, wherein a size of each frame is determined by a coding parameter (Column 3 lines 6-8).

As per dependent claim **7**, McMahon teach, wherein a size of each frame is determined by a page size used by the operating system (Column 5 lines 8-21). *The Examiner notes that the allocator allocates virtual pages that originate from the operating system. Accordingly, the size of each frame is determined by the operating system's page size upon allocation.*

As per dependent claim **8**, McMahon teach, wherein dividing a block of memory into frames includes: determining a last page boundary within the block of memory; and designating a portion of the block of memory after the last page boundary as unused (Column 3 lines 28-37). *The Examiner notes that a free list is present in the system of McMahon. As shown in the table in Column 6 of McMahon, the dynamic memory allocator, in the form of these free lists, catalogs various free blocks of memory by size thereby indicating a boundary for each size that is maintained. Additional memory not fitting into the size constraints would be left as unused.*

As per dependent claim **9**, McMahon teach, wherein a single type of data is stored in the block of memory (Column 3 lines 6-8). *The Examiner notes that the allocator of McMahon allocates memory to requesting programs of the operating system. The system designates each block that is allocated as used after allocation thereby eliminating reallocation of the block to a different program. Accordingly, the system of McMahon allows for a single type of data to be stored in the allocated block of memory.*

As per dependent claim **10**, McMahon teach, wherein data from a fast query system is stored in the instances (Column 3 lines 6-8).

As per independent claim **11**, McMahon teach,

- o an application-level memory manager operable to allocate a block of memory to store data elements, (Column 5 lines 25-27)
- o divide the block of memory into frames, and (Column 4 lines 37-40)
- o divide each frame into instances; and application code operable to define data elements as having an instance type, and to associate the instance type with the instances for storage of the data elements in the instances. (Column 5 lines 50-59).
  - o *The Examiner notes that the dynamic memory allocator takes the blocks (frames) from the whole memory and subsequently divides the blocks (frames) into portions (instances) for allocations. As the request for memory from the application, comes from the application, any allocated memory for that application will contain space available for that application to use. Accordingly, the allocation of memory for the application will contain data storage for the instances using the instances of the requesting application.*

As per dependent claim **12**, McMahon teach, wherein the block of memory begins on a page boundary (Column 3 lines 28-37). *The Examiner notes that a free list is present in the system of McMahon. As shown in the table in Column 6 of McMahon, the dynamic memory allocator, in the form of these free lists, catalogs various free blocks of memory by size. Accordingly, when an allocation takes place by the dynamic*

memory allocator, for example of the 16-byte size, the memory allocated will begin on the 16-byte boundary.

As per dependent claim **13**, McMahon teach, wherein the size of the block of memory is determined by a coding parameter (Column 3 lines 6-8).

As per dependent claim **14**, McMahon teach, wherein the block of memory includes a first page boundary (Column 3 lines 28-37). *The Examiner notes that a free list is present in the system of McMahon. As shown in the table in Column 6 of McMahon, the dynamic memory allocator, in the form of these free lists, catalogs various free blocks of memory by size thereby indicating a boundary for each size that is maintained.*

As per dependent claim **15**, McMahon teach, wherein a portion of the block of memory before the first page boundary is designated as unused (Column 6 lines 34-37). *The Examiner notes that as discussed supra and with respect to the instant citation, the allocator is able to divide the memory block and release the remainder of the block as free space. Accordingly, the allocator designates a portion of the block of memory as unused.*

As per dependent claim **16**, McMahon teach, wherein a size of each frame is determined by a coding parameter (Column 3 lines 6-8).

As per dependent claim **17**, McMahon teach, wherein a size of each frame is determined by the page size used by the operating system (Column 5 lines 8-21). *The Examiner notes that the allocator allocates virtual pages that originate from the*

operating system. Accordingly, the size of each frame is determined by the operating system's page size upon allocation.

As per dependent claim **18**, McMahon teach, wherein the block of memory includes a last page boundary and a portion of the block of memory after the last page boundary is designated as unused (Column 3 lines 28-37). *The Examiner notes that a free list is present in the system of McMahon. As shown in the table in Column 6 of McMahon, the dynamic memory allocator, in the form of these free lists, catalogs various free blocks of memory by size thereby indicating a boundary for each size that is maintained. Additional memory not fitting into the size constraints would be left as unused.*

As per dependent claim **19**, McMahon teach, wherein a single type of data is stored in the block of memory (Column 3 lines 6-8). *The Examiner notes that the allocator of McMahon allocates memory to requesting programs of the operating system. The system designates each block that is allocated as used after allocation thereby eliminating reallocation of the block to a different program. Accordingly, the system of McMahon allows for a single type of data to be stored in the allocated block of memory.*

As per dependent claim **20**, McMahon teach, wherein the application code implements a fast query system (Column 3 lines 6-8).

As per independent claim **21**, McMahon teach,

- o associating data elements used by an application with an application-defined instance type; associating the application-determined instance

type with an application-determined one of a plurality of blocks of memory allocated by an operating system, wherein the application-determined memory block is divided into frames that are further divided into instances; and *The Examiner incorporates herein by reference herein the rejections and citations made supra with respect to claims 1 and 11.*

- populating the instances with the data elements (Column 3 lines 6-8).

As per dependent claim **22**, McMahon teach, wherein associating the application-determined instance type with the application-determined block of memory comprises associating a single application-determined instance type with the application-determined block of memory (Column 3 lines 6-8).

As per dependent claim **23**, McMahon teach, removing the data elements; and returning the block of memory to the operating system (Column 62-65). *The Examiner notes that in the system of McMahon, the realloc() function call allocates a new block of memory and frees the original memory block. This process of allocating a new block teaches the instant limitation of removing the data elements. Once the block is freed, the dynamic memory allocator will find this block upon future searches.*

As per dependent claim **24**, McMahon, returning the block of memory to a buffer; and determining after a predetermined period of time that the block of memory is no longer required by the application (Column 62-65).

Response to Arguments

Applicant's arguments filed 16 March 2006 have been carefully and fully considered but they are not persuasive.

With respect to applicant's argument located within the third full paragraph of the second page of the remarks (numbered as page 8) which recites:

> *"In particular, McMahon is not seen to teach or suggest at least the features of dividing each of the frames into instances and associating an application-defined instance type with the instances."*

The Examiner respectfully disagrees. As taught in McMahon column 5 lines 50-59, the Examiner notes that the dynamic memory allocator of McMahon takes the blocks, 'frames (as instantly claimed)', and maintains a free list indicating each unused bin size, 'instance (as instantly claimed).' Thus the unused bin sizes within the blocks of McMahon, anticipates dividing each of the frames into instances as instantly claimed.

Further, the Examiner notes that the request for memory is received from a software program, 'application.' Upon allocation of memory to the requesting software program, data is stored based on the data type used by the requesting software program, 'application-defined instance type with the instances for data storage (as instantly claimed).'

With respect to applicant's argument located within the fourth full paragraph of the second page of the remarks (numbered as page 8) which recites:

> *"Specifically, although the cited portion of McMahon generally describes the division of memory blocks, nowhere is McMahon seen to describe these divided memory blocks as frames, nor is McMahon seen to divide each of the frames into instances or associate an application defined instance-type with the instances."*

The Examiner respectfully disagrees and refers applicants to the comments made supra with respect to the first point of argument.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Matthew Bradley whose telephone number is (571) 272-8575. The examiner can normally be reached on 6:30-3:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald A. Sparks can be reached on (571) 272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DAS/mb

DONALD SPARKS
SUPERVISORY PATENT EXAMINER